

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-143910

(43) Date of publication of application : 28.05.1999

G06F 17/30  
G06F 13/00

G06F 17/30

G06F 13/00

(71)Applicant : NEC CORP

(72)Inventor : YASUMURA YOSHITAKA

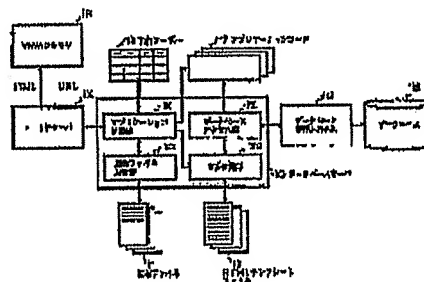
(72)Inventor: YASUMURA YOSHITAKA

(54) EXTENSION SYSTEM FOR COMPOSITE OBJECT AND COMPUTER READABLE RECORDING MEDIUM

(57)Abstract:

**PROBLEM TO BE SOLVED:** To provide technology that is suitable to deal with an object-oriented data base which stores the data of a complicated structure in a composite object form on a WWW(world wide web).

**SOLUTION:** An application code 112 of a server side accesses a data base 105 and returns the object identifier of a base point object of composite object form data coincident with a data base access request. An HTML(hypertext mark-up language) template file 113 defines the type of an attribute value of the object to be actually returned to a user by describing an extension tag to instruct the acquisition of the type of the attribute value to a place where this type is buried. A document extension part 109 accesses the base point object that is pointed by the returned object identifier and another object that is traced from the base point object by a pointer to acquire the attribute value of a necessary object according to the extension tag contained in the file 113. Then, the part 109 converts the acquired attribute value into a text form and generates an HTML document that should be returned to the user.



(11)特許出願公開番号

(43)公開日 平成11年(1999)5月28日

310C  
351G  
380E

(74)代理人 弁理士 境 廣巳

## 【特許請求の範囲】

【請求項1】 インターネットまたはイントラネット経由でクライアント側のWWWブラウザから送出されたデータベースアクセス要求をサーバ側のHTTPサーバ経由でデータベースサーバに伝達し、このデータベースサーバにおいてデータベースアクセス要求で指定されたアプリケーションコードとテンプレートファイルとを使用して、データベースアクセス要求に合致するデータをオブジェクト指向データベースから取得して所定の文書形式に変換し、HTTPサーバ経由でWWWブラウザに返却するWWW-DB連携システムにおいて、

前記アプリケーションコードは、前記データベースをアクセスして利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するように構成されていると共に、前記テンプレートファイルは、実際に利用者に返却すべきオブジェクトの属性値の種類を、それを埋め込む箇所にその取得を指示する拡張タグを記述した構成を有し、かつ、

前記テンプレートファイル中の拡張タグの内容に従って、前記アプリケーションコードから返却されたオブジェクト識別子が指し示す基点オブジェクトおよびこの基点オブジェクトからポインタで辿れる他のオブジェクトをアクセスして必要なオブジェクトの属性値を取得し、テキスト形式に変換して出力文書に埋め込むことで、利用者に返却すべきHTML文書を生成する文書展開部を前記データベースサーバに備えることを特徴とする複合オブジェクト展開方式。

【請求項2】 前記テンプレートファイルは、更に、コレクション名の指定と該指定されたコレクションに含まれる要素を代入する変数の指定と、該変数を使って実際に利用者に返却すべきオブジェクトの属性値の取得を指定した記述とを含むコレクション用の制御構文が、当該属性値を埋め込む箇所に定義された構成を有し、前記文書展開部は、前記制御構文に従って、コレクションに含まれる各要素毎に、その要素を変数に割り当てて実際に利用者に返却すべきオブジェクトの属性値を取得してテキスト形式に変換して順に出力文書に埋め込む構成を有することを特徴とする請求項1記載の複合オブジェクト展開方式。

【請求項3】 前記アプリケーションコードが前記データベースをアクセスして得た前記複合オブジェクト形式データの基点となるオブジェクトの名前とそのオブジェクト識別子とを保持する名前テーブルを備えることを特徴とする請求項2記載の複合オブジェクト展開方式。

【請求項4】 インターネットまたはイントラネット経由でクライアント側のWWWブラウザから送出されたデータベースアクセス要求をサーバ側のHTTPサーバ経由でデータベースサーバに伝達し、このデータベースサーバにおいてデータベースアクセス要求で指定されたア

プリケーションコードとテンプレートファイルとを使用して、データベースアクセス要求に合致するデータをオブジェクト指向データベースから取得して所定の文書形式に変換し、HTTPサーバ経由でWWWブラウザに返却するWWW-DB連携システムにおいて、前記データベースをアクセスして利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するように構成されたアプリケーションコードと、実際に利用者に返却すべきオブジェクトの属性値の種類を、それを埋め込む箇所にその取得を指示する拡張タグを記述した構成のテンプレートファイルと、前記データベースサーバを構成するコンピュータを、前記テンプレートファイル中の拡張タグの内容に従って、前記アプリケーションコードから返却されたオブジェクト識別子が指し示す基点オブジェクトおよびこの基点オブジェクトからポインタで辿れる他のオブジェクトをアクセスして必要なオブジェクトの属性値を取得し、テキスト形式に変換して出力文書に埋め込むことで、利用者に返却すべきHTML文書を生成する文書展開部として機能させるプログラムとを記録したコンピュータ読み取り可能な記録媒体。

【請求項5】 インターネット/イントラネット技術を活用した情報サービス/情報システムの構築において、その系の中で扱われるマルチメディアなどのデータが複合オブジェクトによって構造化された状態でサーバマシン上のデータベースに蓄積されており、利用者がクライアントマシンからデータベースアクセスを要求した際に、その要求事項に応じてどのような処理を行うのかを制御するための手段と、予め提供されているアプリケーションコードに基づいてデータベースにアクセスし、利用者が必要とする複合オブジェクトを取得する手段と、利用者に返却するための出力文書を生成するために、取得した複合オブジェクト内の必要なデータをテキスト形式に変換して出力文書に埋め込む手段から構成されることを特徴とする複合オブジェクト展開方式。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、マルチメディア情報のような複雑な構造を持つデータを複合オブジェクトの形式でデータベースに格納して管理し、WWW(World Wide Web)に代表されるインターネットまたはイントラネット上からそれらの情報を利用者がアクセスするために必要なサーバ側の技術に関し、より詳細にはデータベースから複合オブジェクトを取得した後、それらのデータを、クライアントにいる利用者側に返却する文書形式に変換する際の複合オブジェクト展開方式に関する。

## 【0002】

【従来の技術】インターネット上で情報発信や情報提供

を行うために、WWWが広く利用されている。WWWは、様々な情報が蓄積されているサーバと、利用者が実際に使うクライアントとに分けられる。それらの間はHTTP (HyperText Transfer Protocol) と呼ばれるプロトコルで通信し、クライアントからサーバへはURL (Uniform Resource Locator) により一意に識別される文書や画像などの資源を要求し、その要求に対してサーバからクライアントへHTML (HyperText Markup Language) 文書などの資源を返却する。

【0003】利用者は、クライアント上でWWWブラウザを通してURLの指定やHTML文書の表示などを行い、サーバ上で稼働しているHTMLサーバとのやり取りを行う。WWW上でデータベースへのアクセスを行う場合、基本的にはCGI (Common Gateway Interface) が利用されることが多い。CGIは殆どのHTTPサーバから利用可能なものであり、利用者がサーバ上に予め用意したCGIスクリプトをURLで指定することにより、HTTPサーバがそのCGIスクリプトを別プロセスとして動的に起動して、様々なサービスを提供することができるようになる。CGIスクリプトにデータベースに対する検索要求や更新要求などを記述することで、WWW上のデータベースアクセス処理が実現可能である。

【0004】しかし、CGIスクリプトからHTTPサーバへの返却はHTML文書などの形式にしなければならないため、データベース内のデータ値をテキスト形式に変換したり、適切な場所にタグを埋め込んだりする必要がある。このような処理をCGIスクリプトで実行すると、アプリケーションの作成が容易ではない。そこで、WWWからのデータベースアクセス処理を簡単に記述することを可能にする技術として、WWW-DB連携システムがある。このWWW-DB連携システムもHTTPサーバと連携してCGIを利用するものが多いが、アプリケーション開発者がCGIスクリプト全体を記述する必要はなく、データベースアクセス処理部分とHTML文書への出力のみに着目すれば良いような作成方法になっている。

【0005】WWW-DB連携システムの従来構成の一例を図7に示す。このシステム構成例では、WWWサーバマシン上でHTTPサーバ1002とデータベースサーバ1006とが連携して動作し、関係データベース管理システム1008などをバックエンドとして利用する一般的な構成となっている。

【0006】図7において、利用者はクライアントで動作しているWWWブラウザ1001を利用して、サーバで動作しているHTTPサーバ1002にURLとパラメータ (検索条件など) を渡してデータベースアクセスの要求を出す。HTTPサーバ1002は、このデータ

ベースアクセス要求を処理するためにゲートウェイ部1003にパラメータを渡し、データベースアクセス処理を実際に実行するデータベースサーバ1006に処理を移す。データベースサーバ1006は、予め用意された複数のアプリケーションコード1007のうちのURLで指定されたアプリケーションコードに従ってデータベース管理システム1008を通してデータベース1009にアクセスする。結果はゲートウェイ部1003を通じて文書生成部1004に送られる。

【0007】文書生成部1004は、データベースアクセスの結果を利用者側に返却するためにHTML文書を生成する。図7のシステム構成例では、HTMLテンプレートファイル1005に、生成するHTML文書のひながたが予め作られており、文書生成部1004はデータベースサーバ1006から受け取った結果のデータ値を、HTMLテンプレートファイル1005の指定された位置にテキスト形式に変換して埋め込んでいくことで、利用者に返却するHTML文書を生成する。なお、HTMLテンプレートファイル1005は複数用意されており、何れを使用するかは例えばアプリケーションコード1007中に記述されている。

【0008】文書生成部1004自身はデータベース1009にアクセスしないので、埋め込むべきデータの全てはアプリケーションコード1007の実行時に取得しておく必要がある。文書生成部1004で生成されたHTML文書はゲートウェイ部1003からの出力としてHTTPサーバ1002に渡され、そのままWWWブラウザ1001に返却される。

【0009】また、オブジェクト指向データベース管理システムを利用して、マルチメディアなどの複雑なデータをデータベースに格納してWWW上でサービスするようなシステムも存在する。ただし、システム構成は図7とほぼ同様である。従って、図2に例示するような複合オブジェクト形式のデータがデータベースに格納されており、データベースアクセス要求に合致するデータの各種オブジェクトの属性値を含むHTML文書を利用者に返却する場合、必要な属性値すべてをデータベースから取得するようにアプリケーションコード1007をプログラミングしておき、文書生成部1004は取得された属性値をHTMLテンプレートファイル1005の指定された位置に埋め込む方式となる。

【0010】

【発明が解決しようとする課題】マルチメディアなどのデータは複雑なデータ構造を持っているため、それらのデータを関係データベースに格納するためにはスキーマを構造的に定義しなければならない。従来のWWW-DB連携システムでは、バックエンドのデータベース管理システムとして関係データベース管理システムを利用しているものが多いため、それらのデータへのアクセスには複雑な手順が必要になってしまう。これに対し、関係

データベース管理システムの代わりにオブジェクト指向データベース管理システムを利用すれば、複雑なデータ構造を複合オブジェクトの形式でデータベースに格納することができ、関係データベースを利用するときのような問題は解消される。しかし、WWW-DB連携システムのバックエンドのデータベース管理システムとしてオブジェクト指向データベース管理システムを利用しても、その処理手順が従来のWWW-DB連携システムと同等であれば、以下のような問題が残る。

【0011】(1) アプリケーションコード中のデータベースアクセス処理部分の作成が容易でない。その理由は、利用者に返却すべきオブジェクトの属性値すべてをデータベースから取得するようにアプリケーションコードをプログラミングする必要があるからである。

(2) 利用者に返却すべきオブジェクトの属性値の種類や数の変更時、アプリケーションコード及びHTMLテンプレートファイル双方の変更が必要になる。その理由は、アプリケーションコードを、変更後の属性値を取得するように変更すると共に、この取得した属性値を埋め込む位置を定めるためにHTMLテンプレートファイルを変更する必要があるためである。

(3) 複合オブジェクト形式データに図2に例示するようなコレクションが含まれる場合、HTMLテンプレートファイルの記述が容易でない。その理由は、コレクションに含まれる要素の数だけ、各要素に対応する属性値を埋め込む位置をHTMLテンプレートファイルに定義しておく必要があるからである。

【0012】そこで本発明の第1の目的は、複雑な構造のデータを複合オブジェクトの形式で格納するオブジェクト指向データベースをWWW上で扱うのに好適な技術を提供することにある。

【0013】本発明の第2の目的は、アプリケーションコード中のデータベースアクセス処理部分の作成を容易にすることにある。

【0014】本発明の第3の目的は、利用者に返却すべきオブジェクトの属性値の種類や数の変更時、アプリケーションコードの変更を不要にし、HTMLテンプレートファイルの変更だけで対処し得るようにすることにある。

【0015】本発明の第4の目的は、複合オブジェクト形式データにコレクションが含まれる場合のHTMLテンプレートファイルの記述を簡単にし得るようにすることにある。

【0016】

【課題を解決するための手段】本発明は上記第1乃至第3の目的を達成するために、インターネットまたはイントラネット経由でクライアント側のWWWブラウザから送出されたデータベースアクセス要求をサーバ側のHTTPサーバ経由でデータベースサーバに伝達し、このデータベースサーバにおいてデータベースアクセス要求で

指定されたアプリケーションコードとテンプレートファイルとを使用して、データベースアクセス要求に合致するデータをオブジェクト指向データベースから取得して所定の文書形式に変換し、HTTPサーバ経由でWWWブラウザに返却するWWW-DB連携システムにおいて、前記アプリケーションコードは、前記データベースをアクセスして利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するように構成されていると共に、前記テンプレートファイルは、実際に利用者に返却すべきオブジェクトの属性値の種類を、それを埋め込む箇所にその取得を指示する拡張タグを記述した構成を有し、かつ、前記テンプレートファイル中の拡張タグの内容に従って、前記アプリケーションコードから返却されたオブジェクト識別子が指し示す基点オブジェクトおよびこの基点オブジェクトからポインタで辿れる他のオブジェクトをアクセスして必要なオブジェクトの属性値を取得し、テキスト形式に変換して出力文書に埋め込むことで、利用者に返却すべきHTML文書を生成する文書展開部を前記データベースサーバに備えることを特徴とする。

【0017】このように、アプリケーションコードでは利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するようにし、実際に利用者に返却すべきオブジェクトの属性値の種類は、それを埋め込む箇所にその取得を指示する拡張タグを記述することによってHTMLテンプレートファイルに定義しておき、文書展開部が、HTMLテンプレートファイル中の拡張タグの内容に従って、前記アプリケーションコードから返却されたオブジェクト識別子が指し示す基点オブジェクトおよびこの基点オブジェクトからポインタで辿れる他のオブジェクトをアクセスして必要なオブジェクトの属性値を取得し、テキスト形式に変換して出力文書に埋め込むことで、利用者に返却すべきHTML文書を生成する。

【0018】また本発明は上記第4の目的をも達成するために、コレクション名の指定と該指定されたコレクションに含まれる要素を代入する変数の指定と、該変数を使って実際に利用者に返却すべきオブジェクトの属性値の取得を指定した記述とを含むコレクション用の制御構文をHTMLテンプレートファイルに定義しておき、文書展開部は、前記制御構文に従って、コレクションに含まれる各要素毎に、その要素を変数に割り当てて実際に利用者に返却すべきオブジェクトの属性値を取得してテキスト形式に変換して順に出力文書に埋め込むようにしている。

【0019】

【発明の実施の形態】次に本発明の実施の形態の例について図面を参照して詳細に説明する。

【0020】図1は本発明を適用したWWW-DB連携システムの一実施例のブロック図である。この例のWWW-DB連携システムは、クライアント側に設けられたWWWブラウザ101と、WWWサーバマシン上で稼働しているHTTPサーバ102と、HTTPサーバ102に接続されたデータベースサーバ103と、データベースサーバ103に接続されたデータベース管理システム104と、データベース管理システム104に接続されたデータベース105とから構成されている。なお、WWWブラウザ101とHTTPサーバ102とはインターネットまたはイントラネットにより接続される。

【0021】データベースサーバ103は、名前テーブル110、定義ファイル111、アプリケーションコード112およびHTMLテンプレートファイル113を有し、更に、利用者からのデータベースアクセス要求を受け付け、名前テーブル110を管理してデータベースサーバ内の処理全体を制御するアプリケーション制御部106と、利用者から指定された定義ファイル111を解析し、アプリケーションコード112とHTMLテンプレートファイル113を選択する定義ファイル解析部107と、アプリケーションコード112に記述されているロジック等に従って実際にデータベースへのアクセス処理を行うデータベースアクセス部108と、HTMLテンプレートファイル113に記述された内容により、利用者に返却するHTML文書生成する文書展開部109とを備えている。

【0022】このようなデータベースサーバ103はサーバマシンを構成するコンピュータと記録媒体とによって実現することができる。記録媒体はCD-ROM、磁気ディスク等の機械読み取り可能な記録媒体であり、アプリケーションコード112、定義ファイル111およびHTMLテンプレートファイル113を記録すると共に、データベースサーバ用のプログラムを記録する。ここに記録されたプログラムやデータはサーバマシンを構成するコンピュータに読み取られ、そのコンピュータの動作を制御することにより、そのコンピュータ上に名前テーブル110を含むアプリケーション制御部106、定義ファイル解析部107、データベースアクセス部108および文書展開部109を実現する。

【0023】データベース105はオブジェクト指向データベースであり、データベース管理システム104はオブジェクト指向データベース管理システムである。図2にデータベース105に格納されるデータの例を示す。この例のデータは製品に関する種々の情報を複合オブジェクトの形式で保持する製品データであり、それぞれ1個の製品オブジェクト、外観オブジェクト、拡張オブジェクト、仕様オブジェクトおよび概要オブジェクト、説明オブジェクトと、それぞれ3個の特徴オブジェクトおよびイメージオブジェクトと、1個のコレクションとから成っている。製品オブジェクトは、当該製品デ

ータの基点となるオブジェクトであり、この製品オブジェクトは、当該製品の外観に関する種々の情報を保持する外観オブジェクト、当該製品の仕様に関する種々の情報を保持する仕様オブジェクト、当該製品の拡張に関する情報を保持する拡張オブジェクトおよび当該製品の概要に関する概要オブジェクトへのポインタ（オブジェクト識別子。以下同じ）を保持している。また概要オブジェクトは、当該製品の説明文などを保持する説明オブジェクトおよびコレクションへのポインタを保持し、コレクションは当該製品に関する個々の特徴毎の特徴オブジェクトへのポインタを保持し、各特徴オブジェクトは、その特徴を示すイメージを保持するイメージオブジェクトへのポインタを保持している。データベース105には、このような複合オブジェクト形式の製品データが、異なる製品毎に格納されており、データベース管理システム104を通じて所望の条件に合致する製品データの検索等が可能となっている。

【0024】アプリケーションコード112は、データベース105に対する処理を記述したコードであり、予め定められた処理毎に用意されている。或る1つのアプリケーションコード112ではその中に記述されたメソッド（関数）のロジックによりデータベース105から例えば或る条件を満足するデータを1つ検索する等の処理を実行する。この際、上記メソッドは検索等して得た該当データの基点となるオブジェクトのポインタを検索結果として返却する。例えば、図2に示した製品データを検索対象とする或る1つのアプリケーションコード112中のメソッドでは、例えば与えられたパラメータで指定された製品価格以下の製品データをデータベース管理システム104を通じてデータベース105から検索し、その製品データの基点となるオブジェクト、つまり製品オブジェクトのポインタを返却するようにプログラミングされている。なお、アプリケーションコード112に記述されているメソッドのロジックに従う実際のデータベースへのアクセス処理はデータベースアクセス部108が行う。

【0025】HTMLテンプレートファイル113は、アプリケーションコード112の実行により得られた製品データの情報を要求元の利用者に提示する際、どのような種類の情報をどのような形式（フォーマット）で提示するかを定義したファイルである。文書展開部109はこのHTMLテンプレートファイル113を参照して文書を展開し、利用者に返却すべきHTML文書生成する。HTMLテンプレートファイル113は、提示する情報の種類やフォーマット毎に用意されている。図3に図2の製品データ用に用意されたHTMLテンプレートファイル113の一例を示す。

【0026】図3に示す例のHTMLテンプレートファイル113の記述は基本的にHTML文書の形式に従っており、データベース105から取得したデータを埋め

込むための記法や制御構文をタグを拡張する形で追加している。「\$」が含まれるタグが拡張タグである。このうち、`<$begindetail$>`と`<$enddetail$>`はデータベースアクセスが必要となる箇所をテンプレート中で指定する拡張タグである。また、`<$product.name$>`、`<$product.image.name$>`、`<$product.abstract.text$>`などは、ドット記法によってオブジェクト間の関連やオブジェクトの属性値へのアクセスを指定した拡張タグである。例えば、`<$product.abstract.text$>`は、「productオブジェクトから指されているabstractオブジェクトのtextという属性値を取得する」ということを表している。

【0027】更に、`<$foreach ... から、<$done$>`までの記述は、コレクションの展開のための制御構文（foreach制御構文と呼ぶ）である。一般形式は、`<$foreach (変数名) in (コレクション名) $><$do$>`（HTMLテキスト）…… `<$done$>`という形式であり、（コレクション名）で指定されたコレクションに含まれる各要素毎に、その要素を（変数名）で指定された変数に割り当て、`<$do$>`と`<$done$>`で囲まれた範囲内の（HTMLテキスト）を処理する意味を表している。従って図3の例では、product.abstract.featuresというコレクション名のコレクションに含まれる各要素毎に、その要素をfeatureという変数名の変数に割り当て、featureオブジェクトのtitleという属性値と、featureオブジェクトから指されているimageオブジェクトのnameという属性値とを展開するように指定されている。

【0028】定義ファイル111は、データベース105に対してどのアプリケーションコード112のどのメソッド（関数）を使ってアクセスし、どのHTMLテンプレートファイル113を使ってデータ展開するかを定義したファイルである。つまり、定義ファイル111は以下の情報を保持している。

○アプリケーションコード名

○メソッド名（関数名）

○HTMLテンプレートファイル名

なお、データベース105が複数存在する場合はデータベース名も指定されるが、以下の説明ではデータベース105は1つだけ存在するものとする。

【0029】定義ファイル111は、アクセスの種類、データ展開の仕方等に応じて複数用意されており、データベースアクセス要求時に利用者がURLでその何れか1つを指定できるようになっている。定義ファイル解析部107は、ユーザが指定したURLに従って複数の定義ファイル111の中から1つの定義ファイル111を

選択し、その定義ファイルを解析して、実行すべきアプリケーションコード内のメソッド（関数）、使用するべきHTMLテンプレートファイルを選択する。

【0030】名前テーブル110は、データベースサーバ103内の処理全体を制御するアプリケーション制御部106によって管理されるテーブルであり、ユーザが要求したデータベースアクセス要求の処理に必要な各種の情報を保持するために利用される。代表的なものは、アプリケーションコード112の実行によりデータベース105から取得された複合オブジェクト形式データの基点となるオブジェクトへのポインタである。その他に、データベースアクセス要求時にユーザから指定されたパラメータや、アプリケーションコード112中の実行すべきメソッド名等も保持される。

【0031】次に、上述のように構成された本実施例のWWW-DB連携システムの動作を説明する。

【0032】利用者は、データベース105を利用する場合、クライアントで動作しているWWWブラウザ101を利用して、サーバで動作しているHTTPサーバ102に対し、定義ファイルを指定するURLと若し必要ならデータベースアクセスに必要なパラメータとを渡してデータベースアクセス要求を出し、HTTPサーバ102からHTTP文書が返却されるのを待つ。

【0033】HTTPサーバ102は、データベースアクセス要求を受け取ると、データベースサーバ103にその要求を受け渡して制御をデータベースサーバ103に移す。

【0034】データベースアクセス要求を受けたデータベースサーバ103は、データベース105に格納されているデータへのアクセスが必要になった時点でデータベース管理システム104を通してデータベース105にアクセスし、HTTPサーバ102に返却するための出力文書（HTML文書）を最終的に生成し、HTTPサーバ102に出力する。

【0035】HTTPサーバ102はこのHTTP文書をWWWブラウザ101に出力し、WWWブラウザ101はこのHTTP文書を画面に表示してユーザに提示する。

【0036】本実施例の概略動作は以上の通りであり、次に、本実施例のWWW-DB連携システムの主要部であるデータベースサーバ103の動作を詳細に説明する。

【0037】図4はデータベースサーバ103で利用されるデータの関連を中心にしてデータベースサーバ103の動作を説明した図であり、図5はデータベースサーバ103の全体の処理の流れを示したフローチャートである。

【0038】データベースサーバ103のアプリケーション制御部106は、HTTPサーバ102からURLとパラメータとを指定したデータベースアクセス要求を

渡されると、そのURLを定義ファイル解析部107に渡し、定義ファイル解析部107はURLで指定された定義ファイル111を解析して、その定義ファイル111に記述されたアプリケーションコード名、メソッド名、HTMLテンプレートファイル名を取得し、アプリケーション制御部106に渡す(501)。次にアプリケーション制御部106は、上記取得したメソッド名を名前テーブル110に設定する(502)。図4の例では、名前テーブル110に、queryという名前のメソッドがその種別、型、アプリケーションコード112へのポインタと共に設定されている。

【0039】次にアプリケーション制御部106は、URLと一緒に受け渡されたパラメータを名前テーブル110に設定する(503)。図4の例では、paraという名前を持つパラメータの実際の文字列は名前テーブル110とは別のテキストデータ403の領域に格納し、名前テーブル110には、その名前、種別、型、テキストデータ403へのポインタを設定している。

【0040】次にアプリケーション制御部106は、上記取得したアプリケーションコード名のアプリケーションコード112における上記名前テーブル110に設定したメソッドの実行を開始させる。このメソッドの実行中にデータベースアクセスが必要になると、データベースアクセス部108がデータベース管理システム104を通じてデータベース105をアクセスする。また、データベースアクセスに必要なパラメータはアプリケーション制御部106を介して名前テーブル110から適宜参照される。そして、アプリケーション制御部106は、メソッドの実行により得られた結果データを名前テーブル110に設定する(504)。前述したように、アプリケーションコード112内のメソッドはデータベース105を検索等して得た該当データの基点となるオブジェクトのポインタを検索結果として返却する。つまり、図2に示した製品データを検索対象とするアプリケーションコード112中の或るメソッドでは、例えば与えられたパラメータで指定された製品価格以下の製品データをデータベース管理システム104を通じてデータベース105から検索し、その製品データの基点となるオブジェクトである製品オブジェクトのポインタを返却する。名前テーブル110には、このような製品データの基点となるオブジェクトへのポインタが結果データとして設定される。図4の例では、productという名前のオブジェクトへのポインタが、属性及び型と共に名前テーブル110に設定されている。

【0041】なお、名前テーブル110に設定される結果データ等はアプリケーションコード112の外からも参照できるように、名前テーブル110に設定する前にエクスポートされる。エクスポートされた変数は外部参照が可能になるため、この変数を通してデータベース105内のオブジェクトへのポインタとして名前テーブル

110のエントリに登録することが可能になる。

【0042】アプリケーション制御部106は、結果データの名前テーブル110への設定が終了すると、ステップ501で取得したHTMLテンプレートファイル名を渡して制御を文書展開部109に移す。文書展開部109は、渡された名前のHTMLテンプレートファイルの内容に従って、HTTPサーバ102に返却すべきHTML文書を生成する(505)。HTML文書の生成に際しデータベース105の複合オブジェクトのアクセスが必要となると、文書展開部109は、アプリケーション制御部106を通じて名前テーブル110の結果データを参照して結果データの基点オブジェクトのポインタを取得し、データベースアクセス部108を通じてデータベース105中の基点オブジェクト及びこの基点オブジェクトからポインタで辿れる他のオブジェクトをアクセスする。図4の例には、HTMLテンプレートファイル内の<\$product.name\$>の記述を処理する場合に、productで名前テーブル110を検索してproductオブジェクトへのポインタを取得し、それを使用してproductオブジェクトのnameという属性値をデータベース105から取得している様子が示されている。

【0043】図6は文書展開部109の処理例を示すフローチャートである。以下、図6のフローチャートを参照して文書展開部109の動作を説明する。なお、HTMLテンプレートファイル113中の<\$begindetail\$>と<\$enddetail\$>とで囲まれた部分以外の記述はそのまま出力文書に出力されるため、図6では、<\$begindetail\$>と<\$enddetail\$>とで囲まれた部分についての処理だけを示している。

【0044】文書展開部109は、<\$begindetail\$>を検出すると、まず、foreachのカウンタCを0に設定し(601)、HTMLテンプレートファイル113中の<\$begindetail\$>から<\$enddetail\$>までの記述部分を入力文書C(CはforeachのカウンタCの値。従って最初は入力文書1)として設定し、この入力文書Cに注目する(602)。そして、入力文書Cの先頭から順にトークンの読み込みを開始する(603)。ここでトークンとは、HTMLテンプレートファイル113に記述された字句単位のことであり、<H2>、<\$product.name\$>等がそれぞれ1つのトークンになる。

【0045】入力文書Cにトークンがある場合(603でYES)、それがタグかを判定し(604)、タグでなければ出力文書にそのトークン(固定のテキストなど)を追加し(611)、ステップ603に戻って次のトークンを入力文書Cから読み込む。

【0046】読み込んだトークンがタグであれば(60

4でYES)、それが拡張タグであるかを判定し(605)、拡張タグでなければ出力文書にそのトークン(通常のタグ)を追加し(611)、ステップ603に戻って次のトークンを入力文書Cから読み込む。

【0047】読み込んだトークンが拡張タグであれば(605でYES)、それが制御構文かを判定し(606)、制御構文でなければオブジェクト変数名からの属性値アクセスであるので、ドット表現を辿って属性値を取得してテキスト形式に変換して出力文書に追加する(612)。そして、ステップ603に戻って次のトークンを入力文書Cから読み込む。

【0048】読み込んだ拡張タグが制御構文であれば(606でYES)、その拡張タグがforeachかを判定し(607)、それ以外であれば該当する制御構文に対応した処理を実行し(613)、ステップ603に戻って次のトークンを入力文書Cから読み込む。foreach以外の制御構文としてはif文等が考えられるが、本発明とは直接関係しないので説明は省略する。

【0049】読み込んだ拡張タグがforeachであれば、foreachのカウントCをインクリメントし(608)、foreachに続く変数名の変数に、その直後のコレクション名のコレクションの第1番目の要素を入れ(609)、その直後の<\$do\$>と<\$done\$>とで囲まれたテキスト部分を、入力文書C(従って、入力文書1の次は入力文書2)に設定し、この入力文書Cに注目を移す(610)。そして、ステップ603に戻って、入力文書Cに対して上述した処理を繰り返す。この入力文書Cに対する処理は、<\$do\$>と<\$done\$>とで囲まれたテキスト部分の最初のトークンから最後のトークンまで続けられる。そして、テキスト部分の最後のトークンを処理し終わると、ステップ603で次のトークン無しと判定され、ステップ614へ進む。

【0050】ステップ614では、foreachのカウントが0より大きいかを判定する。0より大きい場合(614でYES)、foreach制御構文の中を処理中であるため、処理中のforeach制御構文で指定されたコレクションに次の要素があるかを調べ(616)、あれば、その要素をforeach制御構文で指定された変数に入れ(617)、次の読み込み先トークンを現在注目中の入力文書Cの先頭のトークンに再設定し(618)、ステップ603に戻って、入力文書Cに対して上述した処理を繰り返す。

【0051】他方、処理中のforeach制御構文で指定されたコレクションに次の要素がない場合(616でNO)、foreachのカウントCをデクリメントし(619)、入力文書C(従って入力文書2の次は前の入力文書1)に注目を移し(620)、ステップ603に戻って、この入力文書Cに対して上述した処理を繰り返す。

【0052】また、ステップ614において、foreachのカウントが0より大きくなければ、<\$begin\$detail\$>と<\$end\$detail\$>とで囲まれた部分についての処理が終了したことになる。その後、出力文書をHTTPサーバ102に返却し(615)、図6の処理を終える。

【0053】図3のHTMLテンプレートファイル例にして文書展開部109の動作を再度説明すると、以下のようなになる。

【0054】foreachのカウントCを0に設定し(601)、HTMLテンプレートファイル113中の<\$begin\$detail\$>から<\$end\$detail\$>までの記述部分を入力文書1として設定し(602)、その先頭から順にトークンの読み込みを開始する(603)。最初のトークン<H2>はステップ611で処理され、出力文書に追加される。次のトークン<\$product.name\$>はステップ612で処理され、データベース105からproductオブジェクトのnameという属性値が取得されて、出力文書に追加される。このとき、基点となるオブジェクト名productをキーに名前テーブル110からproductのエントリが検索され、そこに設定されているproductオブジェクトへのポインタを利用して、データベース105のproductオブジェクトがアクセスされ、nameの属性値が取得される。以降、<UL>までの各トークンも同様に処理される。

【0055】トークン<\$foreach feature in product.abstract.features\$>を読み込んだとき、ステップ608においてforeachのカウントCがインクリメントされて1にされ、次にステップ609においてコレクション名がproduct.abstract.featuresであるコレクションの第1番目の要素が変数featureに入れられる。そして、次のステップ609において、変数featureがコレクションの第1番目の要素であることを前提に、<LI><\$feature.title\$><IMG SRC="<\$feature.image.name\$>">なる入力文書1が処理される。この入力文書1の処理では、先ずトークン<LI>がステップ611で処理され、次のトークン<\$feature.title\$>と<IMG SRC="<\$feature.image.name\$>">とがステップ612で処理される。

【0056】入力文書1の最後のトークン<IMG SRC="<\$feature.image.name\$>">を処理した時点で、ステップ614においてforeachのカウントCが0より大きいことが判定され、ステップ616でコレクションproduct.abstract.featuresに次の要素があることが判定され、その要素が変数featureに設定さ

れ(617)、次の読み込み先トークンを入力文書1の先頭のトークン<LI>に再設定して、再び入力文書1が処理される。この処理では、変数featureがコレクションの第2番目の要素であることを前提に、<LI><\$feature.title\$><IMG SRC="\$feature.image.name\$">となる入力文書1が処理される。以下、コレクションproduct.abstruct.featuresの最後の要素まで同様の処理が繰り返される。

【0057】コレクションの最後の要素まで処理が繰り返されると、ステップ619においてforeachのカウンタCがデクリメントされて0となり、入力文書0に注目を移して処理を続ける。従って、</UL>のトークン以降が処理されることになる。そして、入力文書0の最後のトークン</CENTER>まで処理し終わると、ステップ603で次のトークン無しと判定され、foreachのカウンタCが0なので、ステップ615に進み、出力文書を返却して、処理を終える。

【0058】尚、図3に示したHTMLテンプレートファイルにはforeach制御構文が1つしかないので、foreachのカウンタCは0、1の値しか採らないが、foreach制御構文が入れ子になっている場合には2以上の値をとる。また、図3に示したHTMLテンプレートファイルはアプリケーションコードから1つの結果データのみが返却されることを前提とした例であったが、アプリケーションコードが複数の結果データを返却する場合には、以下のようにすれば良い。先ず、アプリケーションコードでは、各要素に個々の結果データ(例では製品オブジェクトへのポイント)を格納したコレクションを生成し、そのコレクションのポイントを結果データとして返却する。他方、HTMLテンプレートファイルでは、そのコレクションの各要素を順次取り出して図3に例示したような処理を各要素毎に繰り返すように定義しておく。これにより、全製品データについての結果データを一度に返却し、まとめて出力文書を生成することができる。

【0059】

【発明の効果】以上説明したように本発明によれば以下のような効果を得ることができる。

【0060】アプリケーションコード中のデータベースアクセス処理部分の作成が容易になる。その理由は、アプリケーションコードでは利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するようにプログラミングしておけば良く、実際に利用者へ返却すべきオブジェクトの属性値をアクセスする処理

は、全アプリケーションコードに共通な文書展開部が行うからである。

【0061】利用者に返却すべきオブジェクトの属性値の種類や数の変更時、テンプレートファイルの変更だけで対処でき、アプリケーションコードを変更する必要がない。その理由は、アプリケーションコードは利用者からのデータベースアクセス要求に合致する複合オブジェクト形式データの基点となるオブジェクトのオブジェクト識別子を返却するだけで、実際に利用者に返却すべきオブジェクトの属性値の種類はテンプレートファイルの記述に基づき文書展開部が行うからである。

【0062】複合オブジェクト形式データにコレクションが含まれる場合のHTMLテンプレートファイルの記述が簡単になる。その理由は、コレクション用の新規な制御構文を導入したからである。

【図面の簡単な説明】

【図1】本発明を適用したWWW-DB連携システムの一実施例のブロック図である。

【図2】データベースに格納される複合オブジェクト形式データの例を示す図である。

【図3】HTMLテンプレートファイルの記述例を示す図である。

【図4】データベースサーバで利用されるデータの関連を中心にしてデータベースサーバの動作を説明した図である。

【図5】データベースサーバの全体の処理の流れを示したフローチャートである。

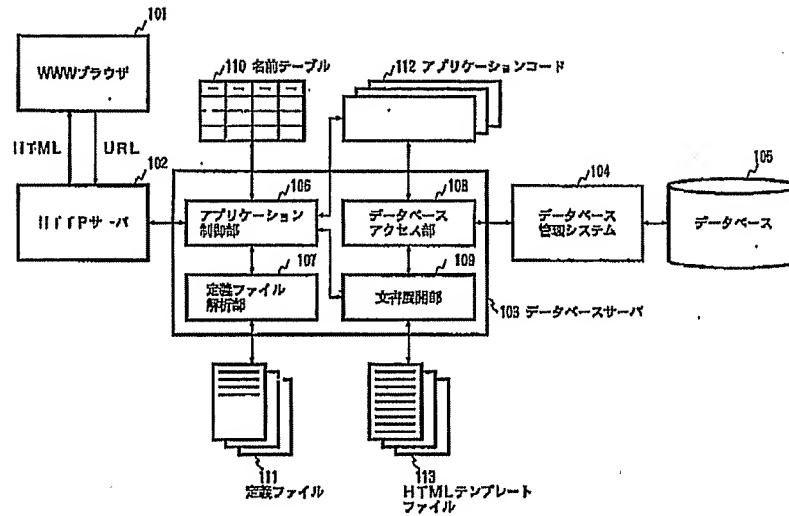
【図6】文書展開部の処理例を示すフローチャートである。

【図7】従来のWWW-DB連携システムのブロック図である。

【符号の説明】

- 101...WWWブラウザ
- 102...HTTPサーバ
- 103...データベースサーバ
- 104...データベース管理システム
- 105...データベース
- 106...アプリケーション制御部
- 107...定義ファイル解析部
- 108...データベースアクセス部
- 109...文書展開部
- 110...名前テーブル
- 111...定義ファイル
- 112...アプリケーションコード
- 113...HTMLテンプレートファイル

【図1】



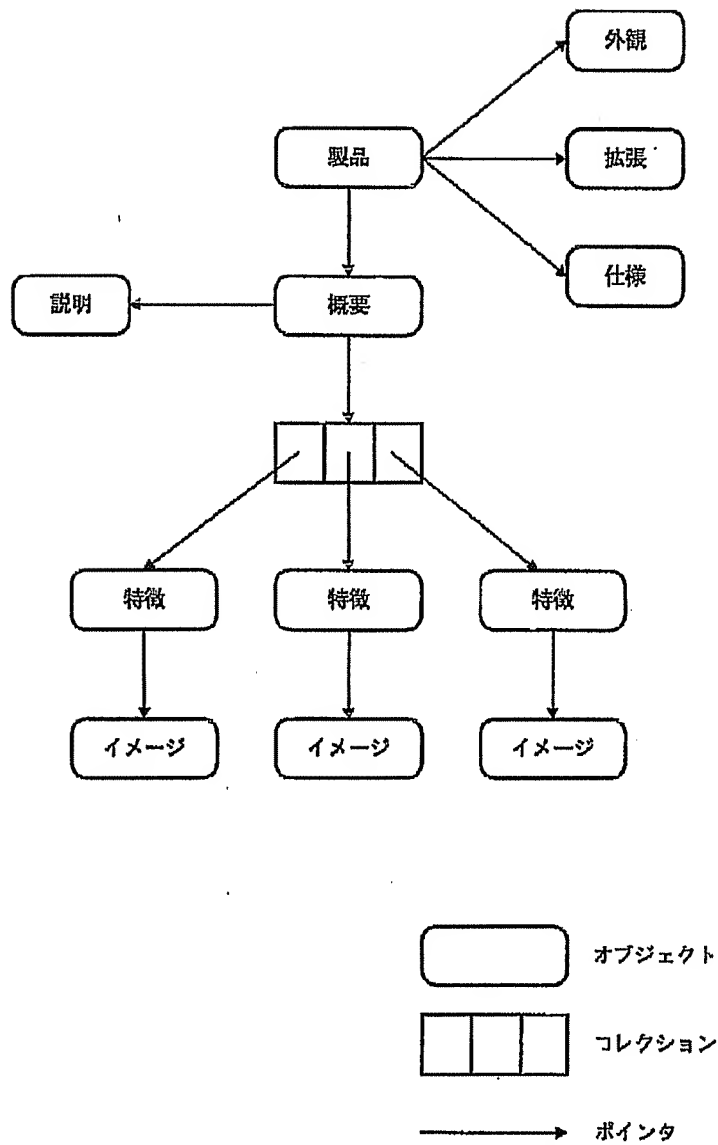
【図3】

```

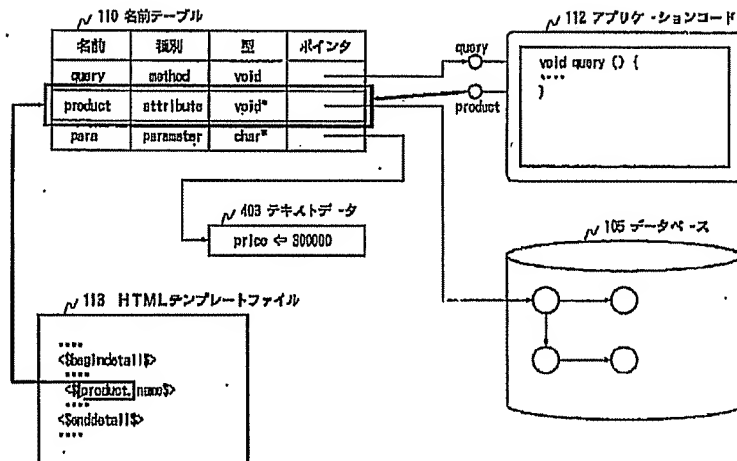
...
<$begindetail$>
<H2><$product.name$></H2>
<IMG SRC=<$product.image.name$>">
<P><$product.abstract.text$>
<P>
<UL>
  <$foreach feature in product.abstract.features$>
  <$do$>
    <LI><$feature.title$>
      <IMG SRC=<$feature.image.name$>">
  <$done$>
</UL>
<CENTER>
  <A HREF=<$product.spec.name$>">仕様</A>
  <A HREF=<$product.extension.name$>">拡張</A>
</CENTER>
<$enddetail$>
...

```

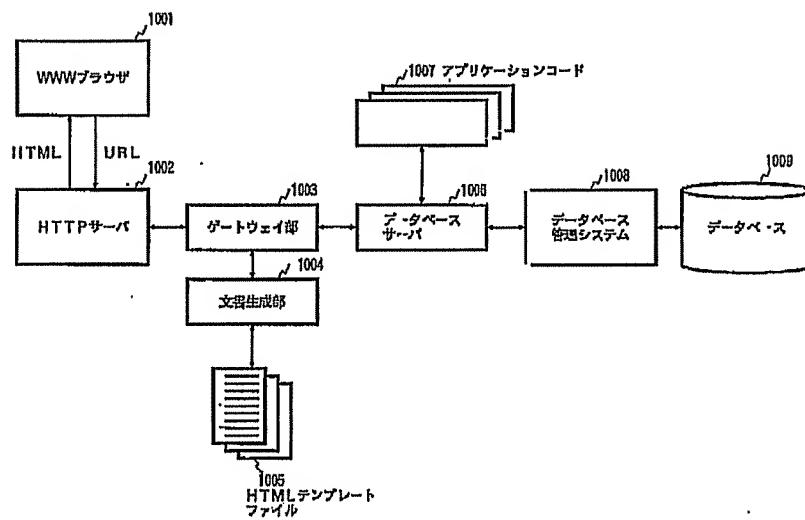
【図2】



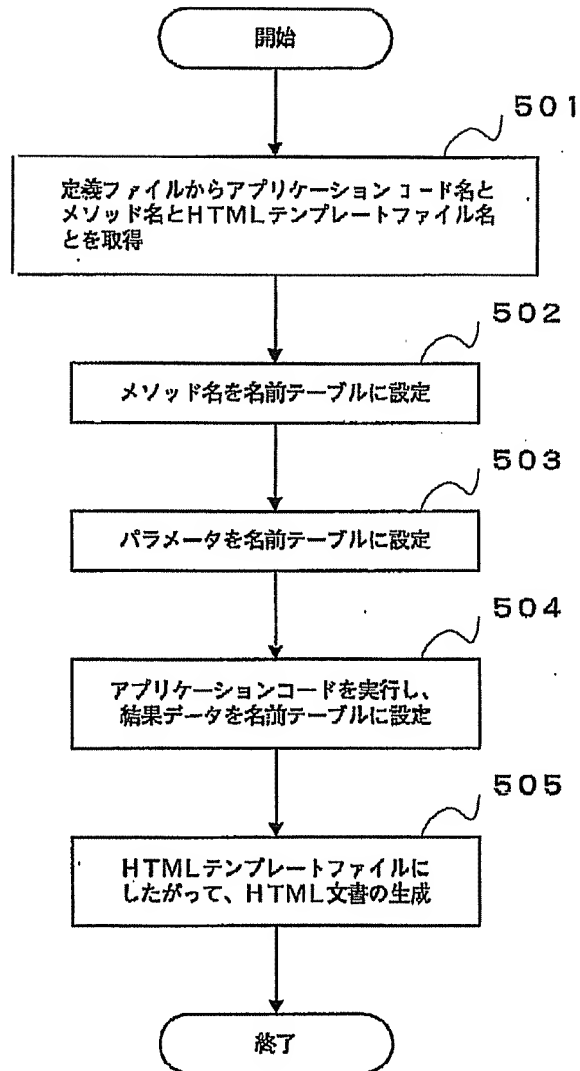
【図4】



【図7】



【図5】



【図6】

